Application of Deep Learning Network in Dog Breed Classification

Rui Xu[1, a], Runlin Liu[2, b]

[1]Department of Computing and Software, Faculty of Engineering, McMaster University, Ontario, L8S 1L9, Canada

[2]Jinan Foreign Language School, Jinan 250108, China

[a]xur52@mcmaster.ca, [b]rainyliurunlin@163.com

Keywords: Dog breed classification; InceptionV3; ResNet50; Deep learning; CNN; Transfer learning

Abstract: With the development of AI technology, machine learning technology has been widely used in image recognition and classification. While most applications are for human features, some advances have been made for animals like dogs. Dogs are common animals in the world, and they play important roles in human societies. Thus, there is a wide range of applications for the use of AI in recognizing and classifying dog breeds. In this paper, we propose two fine-tuned pretrained models (InceptionV3 and ResNet50) to classify dog breeds using convolutional neural network (CNN) with transfer learning. We remove the fully connected layers of InceptionV3 and Resnet50, as well as add optimization layers, so that it is robust and can prevent over-reliance on the classification. The experimental result shows that the fine-tuned InceptionV3 can achieve 89.3% accuracy on the classification of 120 dog breeds, while the fine-tuned ResNet50 can achieve 80.1% accuracy.

1. Introduction

Dogs are ubiquitous animals in human society. According to Fédération Cynologique Internationale (FCI) statistics, there are now 368 officially recognized dog breeds worldwide. Due to the massive number of dogs, the identification and classification of dogs have become very important. The classification of dog breeds plays an essential role in breeding, controlling diseases such as rabies, vaccine development, and customs clearance for pets. However, human beings are very inadequate in recognizing the types of animals, including the breeds of dogs.

Many techniques have been applied in classifying dog breeds. Traditionally, dog breeds can only be well-classified by human experts, but this method is time-consuming, and the number of experts is limited [1]. It has been suggested recently that DNA could be used as an alternative, but this approach is costly and painful for dogs [2]. In 2007, Chanbichitkul et al. [3] proposed a classification method by extracting features of dog faces to construct vectors using both coarse-based classification combined with Principle Component Analysis (PCA)-based classification, and compare these vectors with feature templates in a database of pre-existing breeds. The model achieved a 93% recognition rate across 35 breeds. Wang et al. [4] proposed a method that extracts the facial features of a dog's facial features to build a 2D model that uses SVM regressor and Histogram Intersection Kernel to map features to corresponding categories based on the model. The method can achieve a 96.5% accuracy across 133 different dog breeds. This paper will demonstrate how to use deep learning to recognize and classify 120 different dog breeds.

Deep learning is a specific type of machine learning technology. It is an effective way to realize artificial intelligence and technology that enable computer systems to be improved from experience and data. Additionally, it has strong recognition capabilities and flexibility. Deep learning can discover and express structural features of a given problem to significantly enhance classification performance. It also avoids a series of issues about feature extraction in statistical machine learning [5]. At present, there is a large amount of evidence that in the field of computer vision and image recognition, the recognition accuracy of deep learning is higher than that of traditional image processing and statistical machine learning [6]. A convolutional neural network (CNN) is a deep learning method derived from neural network (NN). In recent years, it has achieved great success in the field of image classification and recognition. Since CNN adopts local connection and weight sharing, it maintains the deep structure of the network, dramatically reducing network parameters so that the model has excellent generalization ability and is easier to train. The problems of network training in NN like vanishing gradient and exploding gradient are solved by CNN [7]. At present, CNN has become one of the research hotspots in many scientific fields. CNN has been widely used in computer vision tasks because it avoids the complex feature extraction of an image and can directly input the original image. It has good adaptability in image recognition and classification, achieving good results in applications in image fields such as handwritten digit recognition and animal classification [8].

The two pretrained CNN models (InceptionV3 and ResNet50) are introduced into dog breed classification in this paper. The rest of the paper is structured as follows: Section 2 contains an explanation of the model as well as the underlying core notion. Then, in Sections 3 and 4, the data study and results are presented. Section 5 concludes with a research summary.

2. Model Explanation

2.1 Structure of CNN

CNN usually consists of a convolutional layer and a sub-sampling layer alternately forming the front end, and the output part adopts a fully connected (FC) layer structure. Figure 1 shows the architecture of CNN.

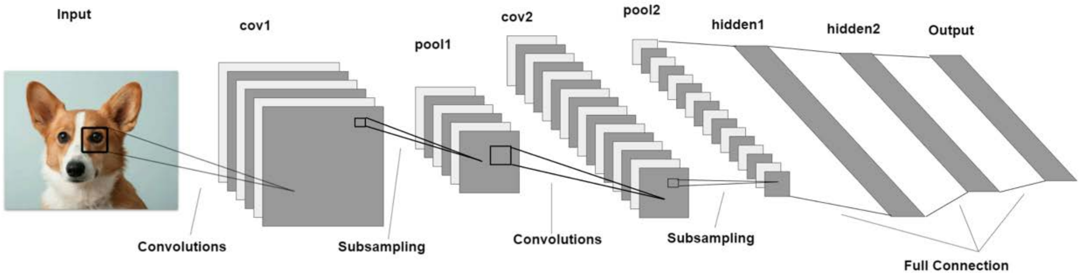


Fig. 1 The overall architecture of Convolutional Neural Network (CNN)

2.1.1 Convolutional Layer

The main task of the convolutional layer is to perform feature extraction, the operation method is convolution operation, and the connection method is a local connection. The convolutional layer is usually composed of multiple feature maps, and they jointly extract features of the previous layer. The connection weights of neurons in each feature map are the same (weight sharing) [9]. In the feature map, each neuron is connected to a neuron (receptive field) in a specific area of the previous feature plane. The convolution operation process first fills in the input feature maps' boundaries to ensure that the feature detector (also known as convolutional kernel or filter) matches the input pixel size and apply the convolutional kernel to the entire feature map. Then the convolution kernel is slid on top of the entire feature map by a fixed stride to construct a new feature map [10]. The connection relationship of the convolutional layer is expressed as:

$$x_i^{n+1} = f\left(\sum x_i^n * w_i^{n+1} + b_i^{n+1}\right)$$

x_i^{n+1} is the output feature of the $(n+1)^{th}$ layer, x_i^n is the output feature of the n^{th} layer, w_i^{n+1} is the weight of the $(n+1)^{th}$ layer, b_i^{n+1} is the bias of the $(n+1)^{th}$ layer, $*$ is convolutional operation, and $f(\cdot)$ is the activation function like sigmoid and relu.

2.1.2 Sub-Sampling Layer

The sub-sampling layer is also called a pooling layer or down-sampling. This layer conducts dimensionality deduction, reducing the number of input parameters and the input image's sensitivity. In the process of convolutional operation in the convolutional layer, multiple feature maps are formed using kernels and to increase dimensionality. Therefore, appropriate dimensionality reduction is required, as otherwise it may cause a dimensionality disaster. The function of the sub-sampling layer is to reduce the dimensionality of the output of the convolutional layer [11]. The neurons on each feature map of the sub-sampling layer are connected to the shared value area of the convolutional layer and the number of feature planes is constant. By mapping the sub-sampling layer to reduce scale, the sub-sampling process can be expressed by:

$$x_i^{n+1} = f\left(pool\left(x_i^n\right) + b_i^{n+1}\right)$$

This is like convolutional layer, x_i^{n+1} is the output feature of the $(n+1)^{th}$ layer, x_i^n is the output feature of the n^{th} layer, b_i^{n+1} is the bias, and $pool(\cdot)$ is sampling function. The process of sub-sampling layer improves the network's ability to resist distortion such as shift and rotation of the input image.

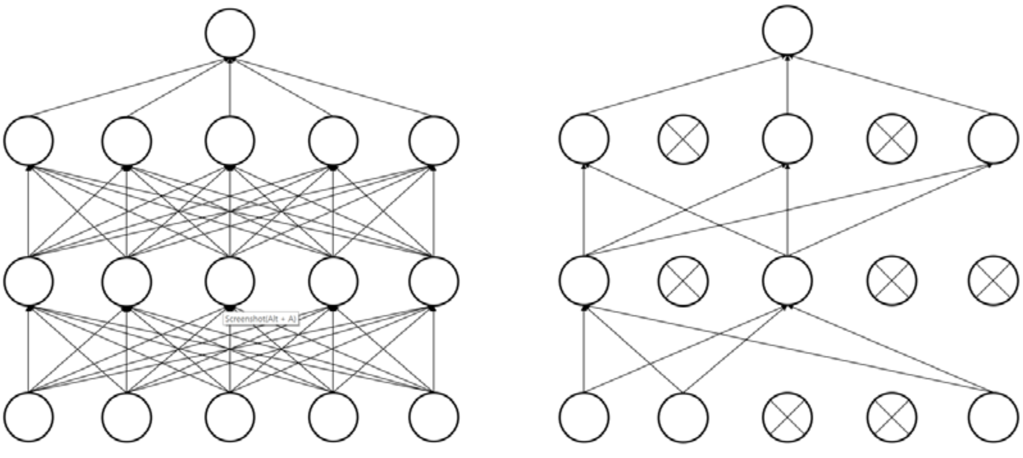
2.1.3 FC Layer

The front end of the CNN is composed of several convolutional layers and sub-sampling layers, and the back-end output part is composed of FC layers. This layer further reduces the dimension of the extracted features and inputs the features into the SoftMax layer.

2.2 Transfer Learning

The CNN model has enormous parameters. One hundred thousand or even one million data is required to train an ideal model for a specific task [12]. In recent years, with the continuous deepening of research, pre-training models have appeared. A pre-trained model is a model trained on an extensive data set (like IMageNet) for a specific task. When solving similar tasks, there is no need to have a massive data set to rebuild the model, we can transfer the pre-trained model to our task and fine-tune the model according to the task. This is known as transfer learning.

Classical CNN usually uses convolutional operations to extract image features in the lower layer, followed by FC layers for classification. However, the number of parameters in the FC layers is enormous, and it leads to two barriers: First, the machine configuration for training is stressful and time-consuming, so reducing the training efficiency is required. Second, it can easily cause overfitting, and not satisfy the generalization ability of the networks. Considering these two issues, we adopt 2 strategies in the experiment when transferring to a pretrained network to decrease the amount of parameters and avoid overfitting: Dropout and Global average pooling.



(a) Standard Neural Network (b) After applying dropout

Fig. 2 Standard Neural Network and Dropout network

Dropout means that the hidden nodes in the deep neural network are randomly dropped in some moments with a certain probability during the training, and the dropped nodes can be considered as temporarily not belonging to the network structure [13]. As shown in Figure 2, the network after Dropout is "thinner" compared with the classical network. The original network has 55 parameters, but the number of parameters reduces to 15 after dropping half of the nodes.

Adding global average pooling layer is another excellent approach to avoid overfitting. Lin [14] proposed an approach to replace FC layers with global average pooling layer, and the experiment shows that global average performs better performance during the training process. As shown in Figure 3, for the convolutional neural network with a FC layer, the output of the feature map corresponding to each set of convolutional kernels is concatenated into the input of the FC layer after using Softmax for multiclassification to obtain the output. For the global level for global average pooling, each feature map is averaged and the average is direct as the input to the Softmax classification corresponding to the output nodes. The global average pooling is used to output only one feature, greatly reducing the parameters while preserving the features.

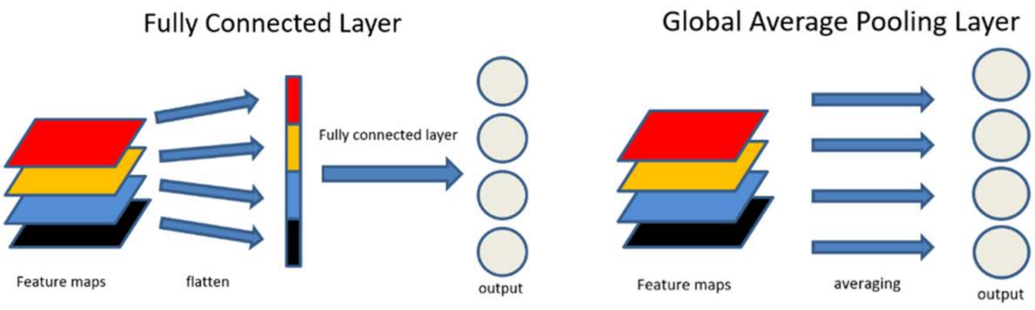


Fig. 3 FC Layer and Global Average pooling

2.3 Pre-trained Models in Experiment

Our fine-tuned models in experiment are based on InceptionV3 and Resnet50.

2.3.1 InceptionV3-based Experimental Model

InceptionV3 is a classic pre-training model in transfer learning. Its network starts with 3 convolutional layers and connect 1 sub-sampling layer. Then, 2 convolutional layers are set up, which are connected to 1 sub-sampling layer. Finally, 11 mixed layers are connected. The original model also has a dropout layer, a FC layer, and a Softmax layer [15]. The overall structure of the classic InceptionV3 network structure is shown in Figure 4.

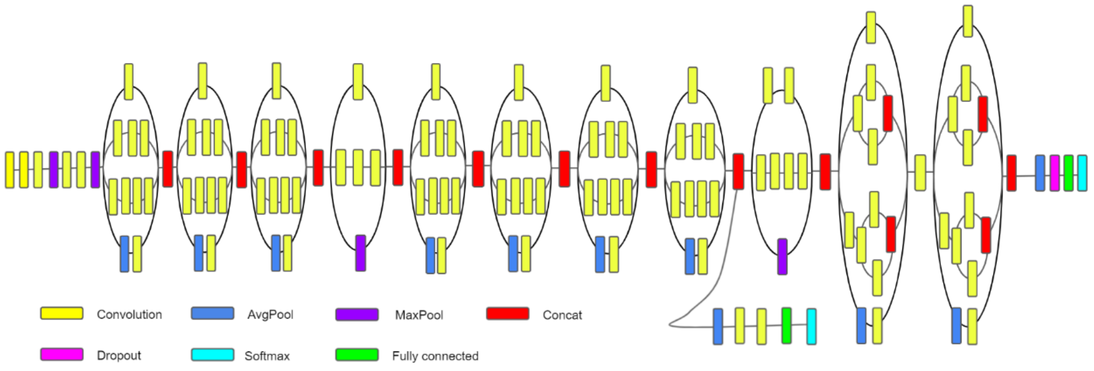


Fig. 4 The overall architecture of InceptionV3

The network structure adopts asymmetric convolution, and it saves a lot of calculation time of parameters on the one and reduces over-fitting. Furthermore, it adds a layer of non-linear extended model expression ability.

The fine-tuned InceptionV3 model proposed in this paper is based on the classic InceptionV3 whose original FC layer is removed, and the parameter optimization layer is added (Dropout and the global average pooling layer). The input of the entire network model is a 299×299 RGB three-channel image. After removing the original FC layer, we add a global average pooling layer and a dropout layer with drop rate of 0.2 before using the Softmax function to connect to the output node. The number of output nodes is thus set to 120 (corresponding to the number of breeds of dogs in the dataset). This structure is shown in table 1.

Table 1 Structure of Fine-tuned InceptionV3

Input (299*299 RGB image)
InceptionV3 with removing FC layer
Global Average Pool
Dropout (0.2)
Dense (Softmax) Output: 120 classes

2.3.2 ResNet50-based Experimental Model

Deep residual network (ResNet) is a deep learning network proposed by Kaiming He. It has excellent performance in target detection and image segmentation. The core of the Resnet network structure is residual learning. While the objective function in a conventional network is the mapping of the optimal solution when solving the parameters of each layer [16]. For the rest network, it does not directly match the optimal solution mapping, but matches 1 residual difference mapping. The architecture of residual network is shown in Figure 5.

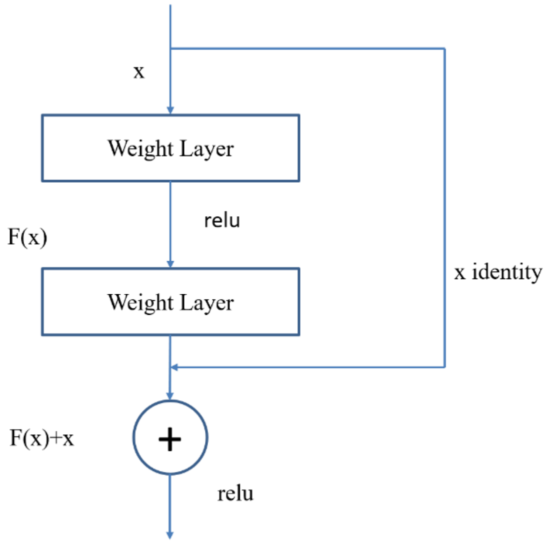


Fig. 5 The Residual Network Architecture

The modification of our dog breed classification ResNet50 model is similar to the InceptionV3 in section 2.3.1. The fine-tuned ResNet50 proposed in this paper is based on ResNet50 with the original FC layer removed. Moreover, the parameter optimization layer is added, using Dropout and the global average pooling layer. The structure is shown in Table 2. The input of the entire network model is a 299×299 RGB 3-channel image. After removing the original FC layer, we add a global average pooling layer with a drop rate of 0.2, and finally use the Softmax function to connect to the output node. The number of output nodes is set to 120. The table 2 shows the structure.

Table 2 Structure of Fine-tuned Resnet50

Input (299*299 RGB image)
ResNet50 with removing FC layer
Global Average Pool
Dropout (0.2)
Dense (Softmax) Output: 120 classes

3. Data Research

3.1 Data Source

The dataset is from “Stanford Dog Dataset”[17]. The dataset includes 120 dog breeds, and there are 20580 images of dogs of different sizes. On average, each breed has 150 - 240 images. The example is shown in Figure 6.

Fig. 6 Stanford Dog Images

3.2 Dataset Augmentation

We expand the original image dataset by 9 times by the means of image zoom, rotation, horizontal flipping, and shifting... finally reaching up to 185,220 images. Figure 7 shows the process.

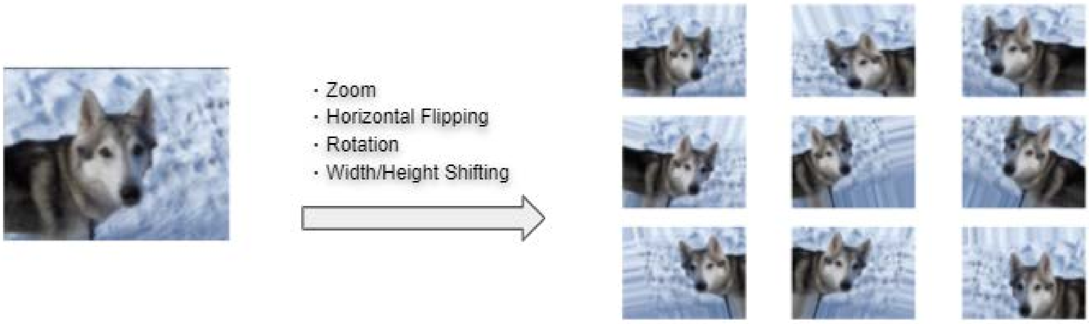


Fig. 7 Dataset Augmentation

3.3 Standardization and Resize

The image sizes in this data set are different, so when constructing the graph data set in the experiment, we first standardize the images and then resize the images to 299*299. The process is shown in Figure 2.

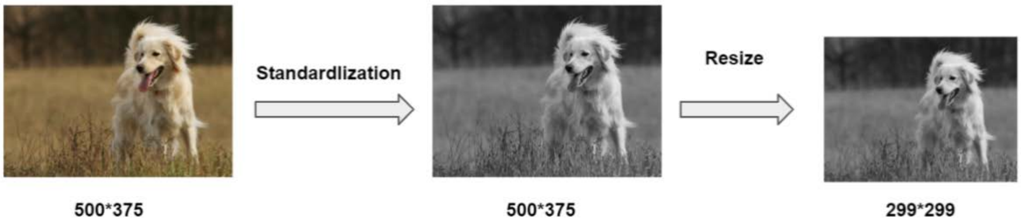


Fig. 8 Standardization and Resize

3.4 The Structure of Data

After preprocessing, the data set is split into a training set composed of 148176 images and a test set composed of 37044 images at a ratio of 8:2. The data structure of the training set and the test set are the same. Take the train set as an example. The data structure of the train set has 2 parts: firstly, the image data set is a 4-dimensional array, and the dimensions are 148176, 299, 299, and 3, respectively, representing a total of 148176 RGB 3-channel images with a pixel size of 299×299. Secondly, the labeled data set consists of a 1-dimensional array, the size of the array is 148176, and each element in the array is a number from 0 to 119, each of which represents 1 among 120 dog breeds.

4. Experiment and Results

4.1 Environment

The running environment of this experiment is the Window10 operating system, in which we select Tensorflow + Keras framework using GPU acceleration during the training process.

4.2 Configuration

In the forward propagation stage, when a dog's image array is input into the network, several convolutional layers and pooling layers extract features, and the resulting features are input to the Softmax classifier, then the corresponding label is used to calculate the loss function. In the backpropagation stage, the weights and biases are updated by layer-by-layer calculation of the partial derivative of the loss function and network parameters. In the training process, the loss function is continuously minimized to improve the classification accuracy until the loss function converges. In this paper, we select fine-tuned InceptionV3 and fine-tuned ResNet50 to classify the breed of dogs and make comparison. In the training process, the same configuration for models training is used, with the input shape for each model set to 299*299, batch size set to 64, and Epoch is set to 15. We select SVG as the optimization function with the learning rate of 0.0001, and the loss function is Cross-Entropy.

4.3 Results and Analysis

After experiment, the training accuracy, validation accuracy, training loss and validation loss of 2 fine-tuned models with the training epochs are shown on Figure 9. The final accuracy and loss results are shown on Table 3.

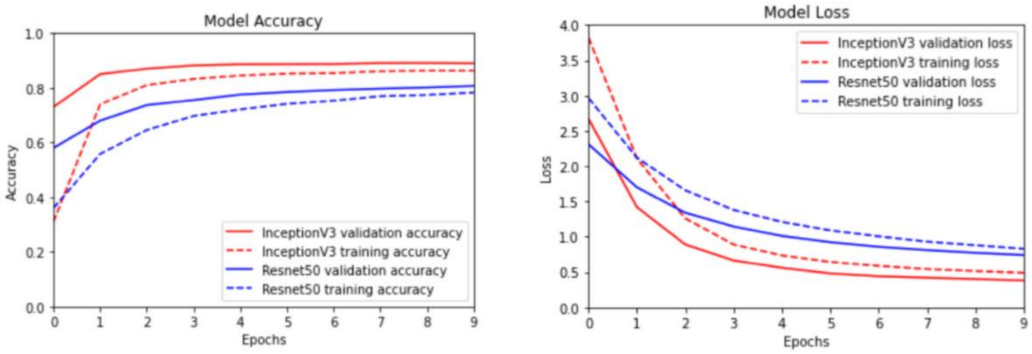


Fig. 9 Accuracy and Loss (ResNet50)

Table 3 Final Experimental Result

Model	Validation Accuracy	Training Accuracy	Validation Loss	Training Loss	Average Time per 1 epoch (second)
Fine-tuned InceptionV3	89.1%	86.2%	0.39	0.48	250
Fine-tuned ResNet50	80.6%	78.3%	0.74	0.83	241

In the experiment, we evaluate the model performance by accuracy and loss. The accuracy is the percentage of correctly classified dog images chosen at random. Cross-entropy is used as the loss function in the neural network to show the training performance. The smaller the value, the better the training model performs. Each training compares the predicted value with the actual value through the cross-entropy function and back-propagates to adjust the weight parameters of the FC layer.

It can be illustrated from Model Accuracy of Figure 9 that the accuracy of 2 fine-tuned models increase rapidly in the first 4 epochs. More specifically, the models tend to converge after 4 epochs, leading to a slight increase in the training accuracy and validation accuracy. Upon completing 10 epochs, the validation accuracy of fine-tuned InceptionV3 and fine-tuned ResNet50 can reach 89.1% and 80.6% respectively.

From the Model Loss graph, the change in the loss of the models correspond to the change in the accuracy of the models. The loss of the models descends rapidly in the first 4 epochs, but the declination becomes slower after 4 epochs. After finishing 10 epochs, the validation loss of fine-tuned InceptionV3 and fine-tuned ResNet50 are 0.39 and 0.74 respectively.

Upon analyzing the result, the training loss and validation loss of fine-tuned InceptionV3 are smaller, and the convergence effect is better than that of fine-tuned ResNet50. This can be also reflected from the accuracy, the validation accuracy of fine-tuned InceptionV3 is 8.5% higher compared to fine-tuned ResNet50. The structures of InceptionV3 and ResNet50 are relatively complex, their average time for 1 epoch are 350s and 355s respectively.

In Summary, while 2 fine-tuned models performed well in the experiment, the fine-tuned InceptionV3 is better.

4.4 Error Samples Analysis

There are still some error samples in the experiment. The reasons for these misidentified breeds are analyzed as follows: First, noises from human faces in some pictures cause the model to have bias in training and prediction. Secondly, some breeds of dogs look very similar, such as Schnauzer and Scottish terriers. To improve upon these defects in the experiment, steps can be taken to increase the number of training samples, as well as preprocess methods such as increasing the accuracy of image segmentation.

5. Conclusion

In this work, we have proposed 2 fine-tuned models based on InceptionV3 and ResNet50 using deep learning to classify 120 dog breeds. With applying transfer learning, the model in the experiments retains most of the parameters of the original model, and we fine-tune the pretrained model to significantly reduce the cost of training. The experimental results demonstrate that both fine-tuned models perform well in classifying dog breeds, with the validation classification accuracy reaching 89.3% for fine-tuned InceptionV3 and 80.6% for fine-tuned ResNet50. These CNN models have excellent robustness and prediction accuracy.

Further improvement could be achieved in the image preprocessing methods and dog face segmentation algorithms to build a better dog breed classifier.

References

[1] Da Fontoura Costa, Luciano, and Roberto Marcond Cesar Jr. Shape analysis and classification: theory and practice. CRC press, 2010.

[2] Voith, Victoria L., et al. "Comparison of visual and DNA breed identification of dogs and inter-observer reliability." American Journal of Sociological Research 3.2 (2013): 17-29.

[3] Chanvichitkul, Massinee, Pinit Kumhom, and Kosin Chamnongthai. "Face recognition based dog breed classification using coarse-to-fine concept and PCA." 2007 Asia-Pacific Conference on Communications. IEEE, 2007.

[4] Wang, Xiaolong, et al. "Dog breed classification via landmarks." 2014 IEEE International Conference on Image Processing (ICIP). IEEE, 2014.

[5] Bengio, Yoshua. "Learning deep architectures for AI." Foundations and Trends in Machinę Learning.-2009.—2 (1).-pp (2007): 1-127.

[6] Russakovsky, Olga, et al. "Imagenet large scale visual recognition challenge." International journal of computer vision 115.3 (2015): 211-252.

[7] Tian, Chunwei, et al. "Enhanced CNN for image denoising." CAAI Transactions on Intelligence Technology 4.1 (2019): 17-23.

[8] Alwzwazy, Haider A., et al. "Handwritten digit recognition using convolutional neural networks." International Journal of Innovative Research in Computer and Communication Engineering 4.2 (2016): 1101-1106.

[9] He, Dongmei, et al. "Vehicle detection and classification based on convolutional neural network." Proceedings of the 7th International Conference on Internet Multimedia Computing and Service. 2015.

[10] Albawi, Saad, Tareq Abed Mohammed, and Saad Al-Zawi. "Understanding of a convolutional neural network." 2017 International Conference on Engineering and Technology (ICET). Ieee, 2017.

[11] Dahl, G. E., Sainath, T. N., & Hinton, G. E. (2013, May). Improving deep neural networks for LVCSR using rectified linear units and dropout. In 2013 IEEE international conference on acoustics, speech and signal processing (pp. 8609-8613). IEEE.

[12] Long, Mingsheng, et al. "Learning transferable features with deep adaptation networks." International conference on machine learning. PMLR, 2015.

[13] Park, Sungheon, and Nojun Kwak. "Analysis on the dropout effect in convolutional neural networks." Asian conference on computer vision. Springer, Cham, 2016.

[14] Lin, Min, Qiang Chen, and Shuicheng Yan. "Network in network." arXiv preprint arXiv:1312.4400 (2013).

[15] Xia, Xiaoling, Cui Xu, and Bing Nan. "Inception-v3 for flower classification." 2017 2nd International Conference on Image, Vision and Computing (ICIVC). IEEE, 2017.

[16] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[17] http://vision.stanford.edu/aditya86/ImageNetDogs